



**CLEAN COPY OF SPECIFICATION
U.S. APPLICATION NO. 10/509,142**



REVIVIFICATION DISPLAY METHOD FOR OUTLINE FONTS BASED ON STROKE CENTERLINES TECHNOLOGY

Field of the Invention

5 This invention relates to data processing, especially relates to the revivification display of data structure font library that could change the data to the data that the computer could process, this invention technology would be used in the Chinese character information processing by computer.

10 Background of the Invention

 The font library and the revivification display of characters are the important contents of the OS. Seeing as the number of the Chinese characters is big, the font library is the base of processing the Chinese information on OS. Using the font library, the Chinese characters could be displayed and imported.

15 There are three conventional font library technologies including bitmap font, vector font and outline font.

 The advantages of bitmap font are that making and displaying of the bitmap font are easily, but it also has its disadvantages that it couldn't be magnified or reduced. That is the magnified and reduced effects are very poor. Besides when it was
20 magnified the font capability would be square increased. Generally, an application need a plurality of fonts having various of sizes, its storage capability requirement is very huge. So that the bitmap fonts were used on the early OS or the embedded equipment that has lower requests on the characters display quality and number of characters.

25 The vector font library and the outline font library are popular on the OS at present, there are two types mostly, TrueType and Postscript. Now the most of PC Os use these two font libraries. The advantages are that these font technologies are mature, the font style is nice and it could be magnified and reduced, so that it can be

used in the PCs. The disadvantages are that the structure of the font library is complex, the font revivification generator is complex, the effect of the small characters display is bad, the revivification speed of characters is slow, the capability of the font library is big and it could not meet the need of the embedded system. The font library technologies were both invented by America. We not only should pay much money, but also the development of our Chinese information processing would be baffled, it is abnormal that we are restricted on the Chinese characters font library technology area.

First of all, in the market there is no outline font library that could meet the need of the embedded system, because the memory capability of the embedded system is small. It will increase the cost while increase the memory capability. One typeface GB18030 of TrueType font takes 20Mbyte storage space, and four typefaces take 80Mbyte in total. At the present the advanced embedded system usually has only 16Mbyte memory. The operating system itself cost some, applications need some, and the left is small and could not hold the huge outline Chinese font library. Many advanced PDAs and cell-phones could not be put in China market because they haven't been Chinese localized. As a result, they haven't found the way to meet the needs of the advanced Chinese embedded system.

Summary of the Invention

The object of this invention is that the font style is nice at every size, the character could be magnified and reduced without restricted, the structure of the font library is legible, the font generator is used easily and the display speed is quick. Besides the font library capability is small so that it could meet the need of the embedded system. It is very necessary to invent a new font library technology.

In addition, we have to invent a new font that is formatted by our own country. This new font library technology should have self-property font style generator, it could be the country standard and it could be the technology barrier when the foreign

company's put embedded products in our market. So it could protect our embedded Chinese information area keeping the top station.

The innovations and difficulties of this invention are that: create a new format and structure font library, this one would accord with the Chinese characters style characteristic and the capability of the font library is small. Our aim is to found the mathematics model to display the Chinese characters using the computer graphics theory and to found the interface that could be called by Chinese OS. According to the exterior data that was called by Chinese OS and combined with the formatted font library data, we could improve the precision and quality of the displaying of the Chinese characters. Then it could get the system support while displaying the characters, that will meet the need of all Chinese information processing, especially it will resolve the problem that the embedded equipment have small memory.

The first one of the innovations of this font library and font generator is the Stroke Centerlines. The conventional font library doesn't describe the Stroke Centerlines, it just describe the whole character outline curve. The Stroke Centerline technology creates the Stroke framework firstly, and then creates the outline based on the centerline. When a character is scaled, only the stroke centerline will change, and the stroke thickness is imported. The subsection of the stroke can describe the different font style. The curvature change could ensure that different size strokes in same kind could be approached. With that, the number of the strokes could be reduced. These all offer the conditions that we could found a structure font library and precise font generator. The capability of conventional font library is larger when they have a good font style. Our inventing technology could make the font library capability smallest with the nice font style, that is, can save 90 percent of the memory capacity. It sounds impossible, but we realize it, because we invent the stroke centerline and the font library with new parameters and structures. Base on the new structure, we invent a new font library generator which could display the characters wonderfully on the computer screen.

The second innovation is that: the conventional TrueType font library could not

display the small size Chinese characters well, so it have to embed bitmap font libraries having several sizes (for example: from 12×12bit to 18×18bit and so on), but this needs more memory capacity. Base on our new font library technology, our font library could display the small size character without embedding small size Bitmap
5 font libraries.

The third innovation is that: the conventional font library technology needs several libraries with different thickness. But our new font technology don't need, we only need a single font library.

The fourth innovation is that: someone tried to use component to combine the
10 whole Chinese character, but they lost. When the same component combines the different characters, the font style will become so bad, because the outline curve coordinate of the component is absolute, when magnifying or reducing the component, the thickness of component will lose the coherence. If the number of components increases, the font library capability will become bigger. Based on our
15 stroke centerline, this problem will be resolved. The strokes have their own thickness, then the component thickness could be changed, the font style would be nice. We could make eligible product.

The invention realization is described as follows: we first found the new parameter and structure of the font library, then found a mathematics model, and
20 realize this on the Windows OS using C language, then on the embedded OS UNLEUS and is solidified on the memory (testing). At last we all hope that it could be made into font library chip and font library explainer chip and solidified into the embedded CPU.

This font library is named as "Revivification display method specialized for
25 outline fonts that based on Stroke Centerlines Technology". It will be called "ZGX" or "ZGX font library generator" for convenience.

Advantages of this technology:

The advantages of this technology include: low memory cost while keeping the

elegance appearance of characters. Usually, one-font standard GB18030 (includes 27590 Chinese characters and 1136 other characters, 28736 in all) will cost 1.1M bytes. Here Standard means every font has its own data file, has specialized parameters, components and strokes. This technology will only need one shared font-
5 lib that takes 800k bytes. Various fonts could share the same data somehow, Song will take 800k, Fangsong, black and Kai only take 600k each and thus in all 2600k (2.6M). TTF will take more than 100M if 7 fonts are needed, in compare, only 4.5M will needed.

This technology is optimized, it could provide in focus characters. It only takes
10 very few memories. It only takes 40k bytes for codes and thus enjoys a high speed. It uses standard C language to program. The Core module uses integers and could be links to various OS. It could change size automatically and add a function to amend the thickness of the strokes that could not be available so far. That would be a crucial advantage for those embedded systems. So it has wide usage.

15

Brief Description of the Drawings

Fig.1 is a flow chart of main module;

Fig.2 is a flow chart of reading in font lib data files;

Fig.3 shows a process of Changing from Standard code/UNICODE to ZGX;

20 Fig.4 is a flow chart of drawing strokes according to ZGX characters' numbers;

Fig.5 is a flow chart of Stroke generator;

Fig.6 shows a flow chart of Bezier lines generator module;

Fig.7 is a flow chart of generating a stroke by one or more Bezier line(s);

Fig.8 is a flow chart of generating a stroke by one or more Beeline(s);

25 Fig.9 is a flow chart of Beeline transfer sub-module;

Fig.10 is a flow chart of Beeline sub-module;

- Fig.11 is Sub-module of drawing bitmaps in buffer;
- Fig.12 is a flow chart of Fill-in sub-module;
- Fig.13 is a flow chart of E table sub-module;
- Fig.14 is a flow chart of CXBOTOOM sort ascending sub module;
- 5 Fig.15 shows file data format of parameterized strokes;
- Fig.16 shows data format of parameterized strokes;
- Fig.17 shows control word format of stroke centerline;
- Fig.18 shows control word format of stroke outline;
- Fig.19 shows file data format of components data;
- 10 Fig.20 shows data format of data file of stroke-composed characters;
- Fig.21 shows data format of stroke structure parameters;
- Fig.22 shows component-composed characters index;
- Fig.23 shows character parameters data format of Component-composed characters;
- 15 Fig.24 shows component parameters data format of component-composed characters;
- Fig.25 shows stroke parameters data format of component-composed characters;
- Fig.26 shows character parameters data format of part shared component-composed characters;
- 20 Fig.27 shows component parameters data format of part shared component-composed characters;
- Fig.28 shows stroke parameters data format of part shared component-composed characters;
- Fig.29 shows a compute screen coordinate system;
- 25 Fig.30 shows a character coordinate system;

Fig.31 shows a character component coordinate system;

Fig.32 shows a stroke coordinate system.

Detail Description of the Preferred Embodiments

To illustrate the operation process with Fig. 1-32: Fig. 1-14 show the flowing charts of all modules consisting of generator. Every module shows how this technology handles with font generation by the algorithm invented according to computer graphic theories. Fig. 15-28 show all data files and data formats needed for the technical solution of the invention and Fig. 29-32 show various coordinates for generating the space by which the data is transformed.

1. Summary

This technology is used for character information processing. It defines a curved outlines font lib data structure. This data structure includes strokes and component data and thus the best characters are generated with generator. It is characterized in that: the lowest memory cost while keeping its elegance. Usually, one-font standard GB18030 (includes 27590 Chinese characters and 1136 other characters, 28726 in all) will cost 1.1M bytes. Here Standard means every font has its own data file, has specialized parameters, components and strokes. This technology will only need one shared font-lib that takes 800k bytes. Various fonts could share the same data somehow, Song will take 800k, Fangsong, black and Kai only take 600k each and thus in all 2600k (2.6M). TTF will take more than 100M if 7 fonts are needed, in compare, only 4.5M will needed. That would be a crucial technology for embedded system, which has limited memory resources. So it has wide usage.

1.1. Strokes of ZGX font libraries

ZGX font lib strokes are the basic elements to form characters. Different strokes are represented by different stroke index. The strokes can be classified as Heng, Shu, Pie, Na, Dian that could be further divided to over 30 groups. In ZGX, over 1000

strokes shall be made. Strokes include Centerlines, constructions and outlines. The outlines will form a closed curve.

The Centerlines are skeletons of the character. It includes control dots for all strokes and consists of beelines and Bezier lines. Strokes include head, body, inflexion and tail. Every part has control dots as reference to describe its outlines. Outlines also consist of beelines and Bezier lines. The stroke sect index indicates its formal index, from 0 to max. The stroke curvature indicates that in the head or tail, some control dots' positions are changed. It has 15 modes from mode 1 to mode 15. The curvature change segment number and the curvature change mode will be given in the index file of stroke library.

The descriptions for stroke centerlines and revivification display generator are shown in Fig. 5-14.

Every font has its own stroke lib.

1.2. ZGX font lib components

ZGX font lib components are formed with some strokes. It could be classified as 1-stroke component, 2-stroke component, 3-stroke component...and the maximum is 31-stroke component. The components are represented by the amount of strokes of component(how many strokes the component has) and their own index number, and such data could be shared around variety-font products.

The strokes forming components have parameters of strokes number, thickness coefficient, center coordinates in x direction and y direction under component coordinate system, scaling coefficients in x direction and y direction and curvature diversification in x direction and y direction.

The stroke number could be shared in various font lib products as common data.

Coordinates for stroke center under component coordinate system are limited from -128 to 127. It is defined when customizing font lib products, and represented in one byte.

The thickness coefficient works on outlines of strokes, it will make strokes thick

or thin. It could make effect on special section and decorates the fonts.

Scaling coefficient in x direction and y direction will only work on control points within centerlines. It will not change the thickness so that the consistency in the thickness of the stroke will be ensured. Scaling coefficient is defined when customizing font lib products, takes a value from 0 to 255 and be saved in one byte also. Wherein 128 denotes double zooming. In compact mode, it could be defined in 6 bits and thus has a limitation from 32-95, the saving format is that the number above described subtracts number 32, that is it is saved as 0 to 63. Whereas 64 denotes double zooming in compact mode.

Curvature diversification in x coordinate and y coordinate means changes in head, tail or some common section. It has 15 modes to work on a single section or 2 sections. (at most 2 sections.) For examples, x coordinates in head and y coordinates in tail is one mode, and x coordinates in common section and y coordinates in it is another mode. The premier will solve compound stroke (such as hengzhe) problem and the ratio problem of heng section and zhe section, and the latter one solve the problem that the same stroke in different component has different curvature diversification. This parameter will make ZGX fonts precisely same as any font templates, so that the precise and beautifulness of the character in font library will be ensured. Each curvature change increment is between -128 and 127, and is expressed with one byte. In the compact font format, the curvature change increment can be expressed with six bits format, because the curvature change increment is usually small.

1.3. Characters of ZGX font library

ZGX font library is expressed with numbers, which can be from 1 to N (N is a long integer), so the amount of the characters is no limited and the font library can be super large. ZGX font library can be ordered according to both Unicode standard and Chinese standard (such as GB2312, GBK and GB18030).

The character of ZGX font library can be formed by several components, also

several strokes directly. The former is called component-composed character, while the later is called stroke-composed character.

Stroke-composed character means that the character is formed by several strokes, whose parameters data include: the number of each stroke composing the character, the thickness of each stroke, the coordinates of each stroke center in the character coordinate system, the scaling coefficients of each stroke in X direction and Y direction, and the curvature change increments of each stroke in X direction and Y direction. The explanation of each parameter of stroke-composed character is the same with the description of components of ZGX font library in section 1.2.

10 Component-composed character means that the character is formed by several components, whose parameters data include: the amount of strokes of each component (how many strokes the component has), the number of each component, the coordinates of each component center under the character coordinate system, the scaling coefficients of each component in X direction and Y direction, and the
15 thickness coefficient of each component. Components are composed of strokes, as explained in section 1.2.

The amounts of each component's strokes (how many strokes the component has) and the numbers of the components are the data that can be shared in multi-font library products. It is only needed to be in the font file of one typeface.

20 The coordinates of each component center under the character coordinate system are decided according to the structure position number. There are 31 kinds of structure position numbers, such as left-right structure, up-down structure, surround structure etc., which are determined when font library product is customized.

The scaling coefficients of component in X and Y directions are limited between 0 and 255, and each is expressed by one byte. In compact format, 0 to 63 represents 32
25 to 95 respectively. Each scaling coefficient is expressed by six bits, and the scaling coefficient "64" means double.

1.4. Extern called parameters for generating character

When a character is generated in the device coordinate system, the parameters used are font number of the character, Chinese standard code (or Unicode standard) of the character, length of the character (unit: pixel), width of the character (unit: pixel),
5 thickness coefficient of the character, foreground color of the character, display mode of the character (such as rotation, bold, oblique, fill etc.), and the bitmap data of the character returning from the buffer.

In principle, the extern called parameters has no limit on the length, width and thickness of the character. This means that one kind of typeface can be any length,
10 width and thickness, so the character can be long or short, bold or thin, also one typeface character may be used as many kinds of typefaces.

1.5. Technical specification of ZGX font library

Under the premise of ensuring preciseness and beautifulness of characters, ZGX
15 characters can save storage and accelerate the display speed mostly.

Standard format means that every font has its independent data files, i.e. different parameter library, component library and stroke library. Compact format means there is a kind of compact expression for data. For example, the parameter originally expressed by 8 bit now is expressed by 6 bit. Multi-font share means that a part of
20 multi-font data is the same to be the shared.

The standard format GB18030 font library (27590 Chinese characters and 1136 non-Chinese characters, 28726 codes together) of one font is 1.1MByte (while TrueType font library is over 15Mbyte). The compact format GB18030 font library of Songti typeface is 800KByte, and the compact format GB18030 font library of
25 Fangsong, Heiti, Kaiti typeface is respectively 600Kbyte. Compact format GB18030 font library of four typefaces is 2.6MByte together, and font library of seven typefaces is 4.5MByte together.

The revivification algorithm of ZGX font library is more excellent than that of

TrueType font library, because the filling process of the former is aimed at each stroke while not the whole character, so the filling speed of ZGX is faster. Additionally, the index of ZGX is simple and direct, and all the math operations are integer operation with overall optimization.

5

1.6 Detail description of revivification display on computer by character library parameter

Hereinafter, the contents in (1) to (5) are description of outline data library, that is, the description of stroke library, stroke-constructed character parameter library, component library, component-constructed character parameter library, character parameter library for multi-font shared format component, and data format. The contents in (6) to (9) are the steps of revivification display on a computer by strokes, stroke-constructed character, stroke-constructed component, component-constructed character. The contents in (10) to (12) are the steps of interface implementing of the font display on computer.

(1) A font library based on the stroke centerline, each font styles have different stroke libraries. The structure of the stroke library is shown in Figures 15-18; every stroke is divided into several segments according to its shape: head, body, corner and tail. There are several bodies and corners in one stroke, the segments are described in lines or Bezier curves. The key points of the segment are the control points of the segments. The relative locations of segments based on the control points describe the coordinates of the outline curve. The control points are the start point and end point of the line, and also they are the start point, outside point and end point of the Bezier curve. The length of the stroke data is not equal to each other. The tail of the stroke ends the stroke.

The data structure of the stroke library is that: the total number of the stroke holds two bytes;

The structure of the head of the stroke is that: the length of the data is equal to the number of the total stroke number multiplying 4. Every stroke holds 4 bytes, the first

3 bytes are the location index of the stroke, the last 4 bits of the last bytes describes the curvature change and the front 4 bits describes the number of the segment that could be curve changed.

The data of the stroke: the data of the stroke centerline and the outline curve data.

5 Different font style have different stroke library, there are 1000 strokes in the stroke library, the data length of the strokes from 20 bytes to 100 bytes.

(2) A font library that is directly constructed by strokes, each font style has different stroke libraries (see Fig.20-21). The structure of the data is:

The relative location of the stroke data holds 4 bytes;

10 The amount of the all characters: 4 bytes;

The index of the characters: the length of the data is equal to the number of character number multiplying 4, and every character holds 4 bytes, the first byte describes the number of its strokes which make up of the character, the next 3 bytes describe the character location.

15 The parameter data: the length is equal to the product of 8 multiplying number of the strokes that make up of the characters, every stroke parameter holds 8 bytes, they are the number of the stroke (2bytes), the thickness (1byte), stroke location (2bytes), scaling coefficient (12bits) and curvature change increment (12bits);

The stroke library: as we have mentioned above in (1).

20 (3). A Chinese component library which are constructed by strokes, each font style has different component libraries (see Figure 19), the structure is that:

The maximum number of the strokes which make up of the components: the total component number (see Figure 19), holds 4 bytes, ranges from 20 to 29, namely the maximum number of the strokes which make up of the components ranges from 20 to

25 29, the arrangement of the components is that: 2-stroke component, 3-stroke component and till to n-stroke component.

The index table of n-stroke components: holds $4*n$ bytes (n is the maximum

number of the strokes which make up of the component), every location number of strokes components holds 4 bytes, that is: the location index of 2-stroke component, the location index of 3-stroke component and till to the location index of n-stroke component, the components which are made up by same amounts of strokes have
5 some, from the number 1 to the maximum number, the components which are made up by 2 strokes hold 2×8 bytes, the components which are made up by 3 strokes hold 3×8 bytes.

The component index data: 8 multiplying the number of the bytes which make up of the component, every stroke index data hold 8 bytes, that is: the stroke number
10 which holds 2 bytes, the thickness coefficient which hold 1 byte, the stroke location which holds 2 bytes, the stroke scaling coefficient which holds 12 bits and the stroke curvature change increment which holds 12 bits.

(4) A font library made up of components and its generator, which are shown in Fig.22-25, each font styles has different parameter data libraries, its data format is
15 that:

Version information: 4 bytes, the front 2 bytes is the name of corporation, the third byte is the symbol of the font style and the last byte is the symbol of the characters sets.

The start location of the character parameter: 4 bytes;

20 The start location of the component parameter: 4 bytes;

The start location of the stroke parameter: 4 bytes;

The character index data table: $(n+1) \times 4$ bytes, the first 4 bytes is the total number, the next $4 \times n$ bytes is the index of every characters, the front 3 bytes is the character location, the last 1 byte is the component number which make up to the character. The
25 character parameter data is that: the component number make up of the character holds 6 bytes, they are that: the component number which holds 11 bits, the strokes number make up of the component holds 5 bits, the component location holds 2 bytes,

the component scaling coefficient holds 12 bits and the thickness coefficient holds 6 bits; the component parameter data: see (3); the stroke parameter data: see (1).

(5) A several font style share structure library, each font style library has the same component number and the same stroke number, these data are described in a basic font and shared by other font, the data structure is that:

The structure of the basic font has been described in (4);

The other font styles structure is that: see Figure 26-28;

The start location of the character parameter: 4 bytes;

The start location of the component parameter: 4 bytes;

10 The start location of the stroke parameter: 4 bytes;

The character index data table: $4*(n+1)$ bytes (n is the total character number), the first 4 bytes denotes the total number, the next $4*n$ bytes denotes the index of every characters, the first 3 bytes denotes the character location, the last 1 byte denotes the component number which make up of the character.

15 The character parameter data is that: the component number make up of the character holds 4 bytes, they are the component parameter which holds 4 bytes, the component location holds 2 bytes, the component scaling coefficient holds 12 bits and the thickness coefficient holds 4bits;

The component parameter data: see (3), while the stroke number is deleted;

20 The stroke library: see (1).

(6) A Stroke centerline technology and method of displaying stroke graph at high resolution in the compute screen coordinate system, see Figure 5-14, Figure 15-18 and Figure 29-32, they are the key and base of the font library which made up by strokes, whether the font library made up by strokes or the font library made up by components, the technology will be boil down to how to display the stroke outline on the computer screen. In order to explain the theory, we would display a character constructed by components at the computer screen coordinate, the mathematics

formula is shown in (6) of section 3.1.

The process of displaying one stroke on the computer screen from the font library constructed by components is that:

5 Firstly, read the parameterized reference stroke data from parameterized stroke library, i.e. the data of drawing centerline segments and outline segments. These data are the coordinates relative to the stroke coordinate system.

10 Because Chinese character's components are composed of strokes, the second step is to transform coordinates of points on strokes from stroke coordinate system into component coordinate system, combine with component and call extern called parameters for data processing at the same time.

15 The Chinese character is composed of several components. Each component has different size and position. The third step is to transform coordinates of composed strokes from component coordinate system into character coordinate system, combine with character and call extern called parameters of the strokes belonging to the component for data processing.

20 Every stroke of Chinese characters will be displayed on computer screen in the end. The fourth step is to transform coordinates of each point of composed strokes from character coordinate system into compute screen coordinate system, and combine with extern called parameters for data processing. These extern called parameters are character's length, width, scaling coefficients, coordinates of character center in screen coordinate system, foreground and background colors for character displaying, and memory array pointer of font's bitmap.

25 Through the data processing and coordinates transform above described, the coordinates of points on stroke centerline segment in screen coordinate system have been gained. Firstly every segment of stroke centerline has been drawn. If the segment has curvature change, then curvature change increment will be added to the corresponding point. And then centerline segments are drawn with lines or Bezier curves according to segment signs. The drawing of lines or Bezier curves use known

algorithms. After drawing all segments, the skeleton line of stroke centerline has been done. Reference to the control points of centerline segments, combining with stroke thickness parameter and curvature parameters of all segments, the coordinates of points of the first outline segments in screen coordinate system can be computed by

5 clockwise from head to tail, and all segments of the first outline of the stroke can be drawn. Then draw the segments of the second outline from tail to head. Coordinates of each outline segment are relative to the corresponding control points of centerline. These relative coordinates multiply thickness coefficients, the extern called parameters, and plus coordinates of corresponding control points of centerline. The

10 results are the coordinates of each outline segment in screen coordinate system. Then outline segments are drawn with lines or Bezier curves according to each segment's signs and drawing mode. After these segments have been drawn, the outline of the stroke is completed. Two outlines form a closed area. Filling this area with known filling algorithms, a stroke of a Chinese character has been displayed.

15 Based on the centerline technology, every stroke would be described precisely, while magnifying and reducing one stroke, it will only act on the centerline control points, and outline will change under the centerline. The change of the character thickness won't change the stroke thickness. That will keep the consistency of strokes thickness of the characters. The thickness parameters only act on the stroke outline. Thus this

20 will realize all kinds of thickness of font styles, especially the thickness is zero, namely only the centerline is drawn. The stroke that only has centerline is nice. The thickness parameters also act on the especial segments, which will keep the font style precise. The curvature parameter could make the font style to match the character models completely. The character models were written by the excellent experts in our

25 country. Our font styles accord with the country character standard. There are 27500 Chinese characters in GB18030, 320000 strokes. We only need 1000 strokes to create our stroke library, based on the parameters we could match the 1000 strokes to the 320000 strokes with different styles, different sizes and different thickness.

(7) A method of displaying the character constructed by strokes, comprising: see

section 4, Figure 4, Figure 20 and Figure 21, Firstly, get the extern called character code, extern called parameters in character revivification array, such as character thickness, character width, character length, character displaying color or grayscale and character display mode. Font size of corresponding font can be got according to the character code index. The amount of components and parameters of each component are read from component-composed font file according to the font size. The parameters of each component are the amount of a component's strokes, component number, coordinates of component center, component scaling coefficients and component thickness coefficients. Every component is displayed after coordinates transforming and date process computing according to the extern called parameters and parameters of every component. For all components of the character, repeat doing according to steps of (6) till all the components are generated on the screen.

(8) A method of displaying the component constructed by strokes, comprising: see section 4.3, Figure 19 and Figure 4. Firstly read the extern called parameters of the character, the parameters include the amount of the component's strokes, the component number, component position, component scaling coefficients, and component thickness. According the component number and this type component index, we could get the strokes data that make up of the component. These stroke parameter data include the stroke number, the stroke thickness coefficient, the stroke location, the stroke scaling coefficient, the stroke curvature change increment, according the coordinate change of the components extern called parameters and the strokes parameters, we could display the every strokes of the components on the screen. When we displayed all the strokes of the character according to steps of (6), we finished the display of the character. There are 27500 Chinese characters in GB18030 standard, and these characters need 5000 components with their parameters.

(9) A method of displaying the characters constructed by components, comprising: see section 4.3, Figure 1 and Figure 22-25, firstly accord to the character coding and the thickness of the characters, the width of the character, the display color, the display mode and the extern called parameters, and the character code

index, we could get the data from the font library, the data include the numbers of the components, the components numbers, the components coordinates, the components scaling coefficients and the components thickness. According the coordinate change of the components' extern called parameters and the character parameters, we could
5 display the every component of the characters on the screen according to steps of (8). When we displayed all the strokes of the character, we finished the display of the character. This font library includes 1000 strokes and 5000 components, these components with their parameters made up of 27000 Chinese characters in GB18030, also these components and strokes could make up of all kinds of font style libraries.

10 (10) A revivification method of High-precision Chinese described by Stroke Centerlines: the method could be applied on various Operation Systems. It is firstly realized on WINDOWS OS in PCs by using standard C language and thus been transplanted to embedded system-NUCLEUS by solidified to memories. Font-lib chips or character Revivification Display chips could be produced by this and then
15 solidified to CPU or embedded OS. The steps for generating high resolving curves by using all kinds of revivification methods have been described in (6)-(9).

(11) An interface for calling Chinese character revivification display by application: also see section 4: ZGX font-lib interface module. The module are consists of 3 sub modules. One init module, also see section 4.2 and Fig 2; One
20 revivification display module, also see section 4.3 and One display device to show character strings, also see section 4.4.

The inputs for init module include: font number that represents Song, Fangsong, Kai and Black in GB18030, both component formed fonts and stroke formed fonts. Even Korean fonts, Japanese fonts or western fonts could be operated. It could handle
25 SBC case and UNICODE/GB2312 cases.

The outputs will return a pointer to arrays. That is the data for all fonts, all components and all strokes. The pointer could be used for revivification display module. Such module will read outer parameters into memory arrays---if such data

are solidify into memories before, they will be seen as constants---those pointer could be read into modules to use and thus ignore the init module.

Inputs for Chinese character revivification display module include: standard code for Chinese characters, Unicode index, font number, out parameters array, return
5 parameter type and pointers to outputs.

Outputs return to computer include: revivification display data according to return type, usually the bitmap data for Chinese characters, a high-speed copied output for such fonts.

The interface will call component from modules (see Fig. 1 and the (9) above
10 described), direct stroke form module (see Fig. 4 and the (7) above described), stroke component sub module (see Fig. 4 and the (8) above described), stroke draw module (see Fig. 5 and the (6) above described) and those parameters obtained from init module.

(12) A device for displaying Chinese strings: see section 4.4. The input is a string,
15 its position is located at up-left corner of the screen, its parameter array (Which has been described in section 4.1, includes outer parameters, width and size, etc)

The outputs are a string showing on screen according to the extern called parameters.

It will be realized by interfaces described in section 4.3.

20 2. Data format of ZGX font library

2.1. Data file of strokes (see Fig. 15)

Data format of strokes (see Fig. 16).

The data format of control word for each stroke center-line segment is represented in one byte. The definition of every bit of the control word is shown in
25 Fig.17.

Data of the stroke center-line segments include: head coordinates (2Byte), control

word of segment (1Byte), body data ($2 \times z00$ Byte), and tail control word.

Data of the stroke outline segments include: control data format of each segment represented in 1Byte, the definition of every bit of the word is explained below (see Fig.18).

- 5 Data format of stroke outline segment: segment control word (1Byte), coordinates increment of 1st edge outline(2Byte), coordinates increment of 2nd edge outline(2Byte), other points coordinates of the head ($2 \times tqy$ Byte), body data ($4 \times z00$ Byte, including points on 1st and 2nd edge outline); corner control word (1Byte), 1st outline coordinates increment (2Byte), 2nd outline coordinates increment (2Byte),
10 other points coordinates of the corner ($2 \times tqy$ Byte), body data ($4 \times z00$ Byte, including coordinates of points on 1st and 2nd outline); tail control word, other points coordinates of the tail ($2 \times tqy$ Byte).

2.2. Data file of components

- 15 Please see Fig.19.

2.3. Data file of Stroke-composed characters

The first four bytes is the start position of stroke data (with regard to head file).

The next four bytes is the total number of Chinese characters.

- 20 Then, the index data of every Chinese character takes four bytes orderly, where the first one byte is the stroke amount of the character, and the next three bytes are the position of the stroke data in the stroke parameter file (see Fig.20).

The length of the data block is the product of 4 multiplying the total number of Chinese character.

- 25 The following data block is the stroke structure parameters: every stroke takes 8 bytes, where the first two bytes is the stroke number, the third byte is the thickness of the stroke, the next five bytes are the X, Y, SX, SY, QX, QY of the stroke (see

Fig.21).

The storage of the data block is: amount of all Chinese characters' strokes*8 bytes;

Next, it is the stroke generating data block. It is composed of:

5 The amount of strokes: 2 bytes;

The index of stroke data: 4*amount of strokes, stroke data (Fig.15 is the compact format of stroke data file).

2.4. Data format of Component-composed characters

10 Basic font file is mainly composed of several parts: index table, character parameter data, component parameter data and stroke data.

Index table (see Fig.22)

Character parameter data (see Fig.23)

Component parameter data (see Fig.24)

15 Stroke data (see Fig.25)

2.5. Format of part shared multi-font component-composed characters

Other font files are mainly composed of index table, character parameter data, component parameter data, and stroke data. The data formats in detail are described as

20 following:

Index table (see Fig.22)

Character parameter data (see Fig.26)

Component parameter data (see Fig.27)

Stroke data (see Fig.28)

3. The mathematical model and its font generator device

3.1. Coordinate system and transforming formula

ZGX font library uses following coordinate systems: device coordinate system (relative to printer or other devices), character coordinate system, component coordinate system and stroke coordinate system.

(1) Device coordinate system

Take computer's monitor as an example, the definition of the screen is as Fig.29.

The origin of the screen is (0,0) at the top left corner. The Y-axis is down, and the X-axis is rightward. The resolution of VGA is 640*480, so the coordinates of the bottom right corner is (639,479).

(2) Character coordinate system

In Fig.30, $x_1o_1y_1$ is the coordinate system, and xoy is the device coordinate system. The center of the character is in the device coordinate system. To display a character, the following data are needed: the number of the character, the scaling coefficients in X direction and Y direction (the resolution of the reference character is 256*256) and the coordinates of the character's center $o_1(x_0, y_0)$ (this point is the origin o_1 of the character coordinate system's center). Then the character can be displayed right in the given position.

(3) Component coordinate system

In Fig. 31, $x_2o_2y_2$ is the component coordinate system, relative to character coordinate system. And o_2 is the center of a component. Every component has a

component coordinate system relative to character coordinate system. Displaying a character is realized by displaying several components. To display a character, it is needed to read the number of each component, the scaling coefficients in X and Y directions, and the relative coordinates (x_{10}, y_{10}) of o_2 .

5 (4) Stroke coordinate system

In Fig.32, $x_3o_3y_3$ is the stroke coordinate system, relative to component coordinate system $x_2o_2y_2$. Generating a component is realized by displaying several strokes. To generate a component, it is needed to read number of each stroke, the scaling coefficients in X and Y directions, and the relative coordinates (x_{20}, y_{20}) of o_3

10 in $x_2o_2y_2$ coordinate system. Also, every stroke has a stroke coordinate system.

How to generate a stroke? In the stroke coordinate system, the stroke is gained by filling algorithm according to the stroke's coordinates and curve fitting parameters (read from stroke file according to the number of the stroke).

(5) Fitting of second-order Bezier curve

15 It is shown in Fig.32. The short Pie is divided into three segments, and every segment uses a fitting of second-order Bezier curve.

The first segment uses the coordinates of ①, ②, and ③. The ② is the point of intersection of the tangents of ① and ③.

The second segment uses the coordinates of ③, ④, and ⑤. The ④ is the point of
20 intersection of the tangents of ③ and ⑤.

The second segment uses the coordinates of ⑤, ⑥, and ①. The ⑥ is the point of intersection of the tangents of ⑤ and ①.

If curve from ① to ③ is a line, than ② can be any point on the line.

The formula is as following:

$$\begin{aligned} x(t) &= x_1(1-t^2) + 2t(1-t)x_2 + t^2x_3 & (1) \\ y(t) &= y_1(1-t^2) + 2t(1-t)y_2 + t^2y_3 & (2) \end{aligned} \quad \text{wherein, } t \in [0,1]$$

In the formula, (x_1, y_1) , (x_2, y_2) and (x_3, y_3) are respective the coordinates of the head point, the middle point and the end point.

- 5 Thus, it is only needed to save the relative coordinates of ① to ⑥ points for Bezier curve fitting. The data format is the same as that of broken line, but the difference is the record of coordinates of even points. And the point is the intersection of the tangents of one-and-aft two points.

(6) Coordinates transform of generating a character in a device

- 10 The transform formulas for coordinate points on every stroke of every component from character coordinate system to device coordinate system are:

$$X_{kij} = ((X_{2i} \quad S_{xBi} \quad X_{3j} \quad 128) \quad B_{xk} \quad 128 \quad X_{k1}) \quad Z_{xs} \quad 128 \quad X_0$$

$$Y_{kij} = ((Y_{2i} \quad S_{yBi} \quad Y_{3j} \quad 128) \quad B_{yk} \quad 128 \quad Y_{k1}) \quad Z_{ys} \quad 128 \quad Y_0$$

Where,

- 15 X_{kij}, Y_{kij} denotes coordinates of the i th stroke in the character coordinate system;

Z_{xs}, Z_{ys} denotes scaling coefficients of the character;

X_0, Y_0 denotes character's coordinates in the screen coordinate system;

X_{3j}, Y_{3j} denotes the i th stroke's coordinates in the stroke coordinate system;

X_{k1}, Y_{k1} denotes coordinates of the k th component's center in the character

coordinate system;

B_{xk}, B_{yk} denotes scaling coefficients of the k th component's center in X and Y directions;

X_{2i}, Y_{2i} denotes coordinates of the k th component's i th stroke's center in the character coordinate system;

S_{xBi}, S_{yBi} denotes scaling coefficients of the k th component's i th stroke in X and Y directions;

$k=1$ to the amount of components of the character;

$i=1$ to the amount of strokes of the k th component;

$j=1$ to the amount of points of the k th component's i th stroke;

in case of $k=1$, the character is composed of strokes directly, and the formula above described is used for the coordinate transform when constructing character by strokes.

3.2. ZGX font generator device

The flow chart of the module is as following:

- (1) The flow chart of main module, as shown in Steps 110-120 of Fig.1.
- (2) Read font library data into memory array, as shown in Steps 210-220 of Fig.2.
- (3) Transform Chinese Standard code (or Unicode code) into character number of ZGX font library, as shown in Steps 310-316 of Fig.3.
- (4) Generate Stroke-composed character according to the character number of ZGX font library, as shown in Steps 410-414 of Fig.4.
- (5) Generate Component-composed character according to the character number of ZGX font library, as shown in Fig.1.
- (6) The flow chart of stroke generator, as shown in Steps 510-524 of Fig.5.
- (7) Recursive call module of Bezier curve, as shown in Steps 610-614 of Fig.6.

(8) The sub-module of generating a stroke segment with one or more Bezier curve is shown in Steps 710-711 of Fig.7.

(9) The sub-module of generating a stroke segment with one line or more, as shown in Steps 810-811 of Fig.8.

5 (10) The sub-module of transforming line, as shown in Steps 910-911 of Fig.9.

(11) The sub-module of drawing line, as shown in Steps 1010-1017 of Fig.10.

(12) The sub-module of drawing bitmap in buffer, as shown in Steps 1110-1111 of Fig.11.

(13) The sub-module of filling algorithm, as shown in Steps 1210-1215 of Fig.12.

10 (14) The sub-module of edge table (ET) of edges, as shown in Steps 1310-1316 of Fig.13.

(15) Insert all edges of one item from ET of edges into active edge table (AEL) in ascending order by cxBottom, as shown in Steps 1410-1416 of Fig.14.

15 4. ZGX font library call interface device

4.1. Data structure definition

(1) Return pointer array of font initialization

```
typedef struct fhzz
```

```
{
```

20 long * zys;

```
long * zics;
```

```
long * zrcs;
```

```
long * bhcs;
```

```
}fhzz, far * lpfhzz;
```

25 zys: data pointer of character index parameter;

zics: data pointer of character parameter for font;

zrcs: data pointer of component parameter for font;

zhcs data pointer of stroke parameter for font.

(2) Revivification array of (Chinese) character

5 typedef struct hzfycs

{

unsigned char zhd;

unsigned char zkd;

unsigned char zcd;

10 unsigned char zys;

unsigned char zfs;

} hzfycs, far * lphzfycs;

zhd: the thickness parameter of Chinese character, 1 Byte, its standard value is 32;

zkd: the width parameter of Chinese character, 1 Byte, using pixel as unit;

15 zcd: the length parameter of Chinese character, 1 Byte, using pixel as unit;

zys: the color or grayscale of Chinese character, 1 Byte;

zfs: the display mode parameter of Chinese character, 1 Byte.

The definition of parameter value is as following:

The first two bits denote the rotation mode: 00 means the routine (no rotation), 01
20 means 90 degree rotation, 10 means 180 degree rotation, 11 means 270 degree rotation.

The third bit denotes the bold or not: 0 means routine, 1 means bold.

The fourth bit denotes the italic or not: 0 means routine, 1 means italic.

The fifth bit denotes that it is underlined or not: 0 means routine, 1 means underline.

The sixth and seventh bits denote the filling mode of Chinese character: 00 means only drawing centerline i.e. single line character, 01 means filling Chinese character by 'OR' mode, 10 means filling Chinese character by 'AND' mode, 11 means filling Chinese character by 'XOR' mode.

The eighth bit denotes the display mode of character: 0 means DBC case, 1 means SBC case, 1 means Chinese character.

10 4.2. Initialization of font

(1) Interface module

HZ_S (zth, lpfhzz);

zth: font size, character parameter, these are entrance parameters;

The definitions of parameters are as following:

15 0: 16*16 bitmap font library according to GB-2312 standard;

1: Outline font library of Songti (Chinese) font according to GB18030-2000 standard;

2: Outline font library of Heiti (Chinese) font according to GB18030-2000 standard;

20 3: Outline font library of Fangsongti (Chinese) font according to GB18030-2000 standard;

4: Outline font library of Kaiti (Chinese) font according to GB18030-2000 standard;

5: Outline font library of special characters according to GB18030-2000 standard;

25 6: ...

Lpfhzz: return parameter, pointer of pointer array.

(2) Realization method

Open file according to the size of font, for example, the name of GB2312 16*16 font library file is DW16D, the name of GB18030-2000 Songti outline font library file is DWSTQ, the name of GB-2312 Heiti outline font library file is DWHTQ, and the name of special character font library file is DWQZFK.

Twelve memory array are defined respectively as GB-2312 16*16 bitmap font array, index parameter array of GB18030-2000 Songti outline font library, character parameter array of Songti font, Songti component parameter array and Songti stroke parameter array; character index parameter array of Heiti outline font library, character parameter array of Heiti font, Heiti component parameter array and Heiti stroke parameter array; parameter arrays of Fangsong, Kaiti, and special character etc.

Read file data according to the font into corresponding array;

Return the pointer of each array.

4.3. Revivification of Chinese characters

(1) Interface module

HZ_X(hz\$,zth,hzfycs,fclx,zbuf);

Hzh: size of font;

hz\$: the Chinese standard code of a Chinese character, usually 2 bytes;

hzfycs: the revivification parameter array of Chinese character, 5 bytes;

Fclx: the type of a return parameter, 0 denotes bitmap, 1 denotes outline or vector array, 2 denotes TTF data format, 3 denotes PS data format;

Zbuf: bitmap of returned Chinese character or other data buffer pointer; this is a returned parameter.

The storage mode of Chinese character bitmap is that every pixel takes 1 bit, every character takes $\lceil ((zkd * zcd) + 7) / 8 \rceil$ bytes.

(2) Realization method

Find the corresponding array pointer according to the size of font;

Calculate the size of Chinese characters according to the inner code. The formula for calculation is as following:

Suppose the higher byte of the inner code is C1, and the lower byte is C2 (C1, C2
5 are the integer with no sign). If C1=0, then go to DBC case process.

If hz\$ is in the range of character code, i.e. the Chinese standard code is between
0X1F and 0X80, A1A1-A1Fe and A9A1-A9FE, A840-A87E and A880-A8FE and
A940-A97E and A980-A9FE (total amount is 1136), the character font library
DWQZFK will be called. Outline font library uses two steps to generate Stroke-
10 composed characters. The module of stroke revivification needn't to be modified. It is
only needed to modify the called outline font library file, process the structure file of
the character, and calculate the order number ZH of the character in the font library
according to the code:

```
if(hz$>=0x1f && hz$<=0x80) ZH=hz$-0x1f;  
15 else if(c0>=0xa1 && c0<=0xa9 && c1>=0xa1 && c1<=0xfe)  
  
    ZH=99+(c0-0xa1)*94+c1-0xa1;  
  
    else if(c0>=0xa8 && c0<=0xa9 && c1>=0x40 && c1<=0xa0)  
  
    {  
  
        ZH=945+(c0-0xa8)*96+c1-0x40;  
  
20 if(c1>0x7f) ZH--;  
  
    }
```

Else go to the corresponding font file for Chinese font library, and do the following process:

If C1>=0XB0 and C1<=0XF7 and C2>=0XA1, then character number =(C1-
25 176)*94+C2-161;

If C1>=0X81 and C2<=0XA1: if C2>=0X7F, then C2--, character number =(C1-

176)*94+C2-64+6768;

If $C1 \geq 0XAA$ and $C2 \leq 0XA1$: if $C2 \geq 0X7F$, then $C2--$, character number = $(C1 - 170) * 96 + C2 - 64 + 12848$;

The number of four-byte character is between 21008 and 27589.

5 The data describing the character are found according to the character number;

If the character is a 16*16 bitmap font character, then the data pointer of the character is: bitmap font data pointer + (character number - 1) * 32;

If the character is an outline font character, then its character parameter data will be found according to character index file and character number, then the component
10 parameter data will be found according to the character parameter data, and finally the stroke data will be found according to each component parameter data.

The coordinates of each stroke are calculated.

The drawing stroke sub-module is called.

The drawing stroke sub-module calls drawing line sub-module and drawing Bezier
15 curve sub-module. In the end, the filling module is called.

Additionally, there are rotation, bold, underline, and italic modules.

(3) Design

An interface module, a drawing stroke sub-module, a drawing line sub-module, a drawing Bezier curve module and a filling sub-module are required by design.

20 The stroke drawing line sub-module and the filling sub-module need to be realized in buffer.

The calculating of the character drawing control is on the 256*256 rectangle. It is scaled when drawing centerline and outline (firstly, both the length and width of the character are 256; when drawing centerline and outline of the stroke, the following
25 calculating is applied: X coordinate relative to the character's center * character's width / 256, Y coordinate relative to the character's center * character's length / 256).

Return the bitmap, vector outline, TTF format or PS format of the character.

Fault tolerance: All characters can be displayed at any resolution without error, distortion and beyond the limits.

4.4. Display of character string

5 (1) Interface module

HZ_C(hzc\$,zth,zxx,zyy,hzfycs);

Hzh: size of font;

hzc\$: the code string of a Chinese character or a set of characters;

Zxx: the X coordinate of the displaying character string in the top left corner;

10 Zyy: the Y coordinate of the displaying character string in the top left corner;

hzfycs: the revivification parameter array of characters, 5 bytes.